

Opinnäytetyö (AMK)

Tietojenkäsittely

Sähköisen liiketoiminnan järjestelmät

2013

Alexi Älli

# WORDPRESS-TEEMAN SUUNNITTELU JA TOTEUTUS

– Case: Puhelinklinikka.fi



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Titetojen käsittely | Sähköisen liiketoiminnan järjestelmät

Joulukuu 2013 | 25 sivua

Ohjaaja: Minna Paakki

Alexi Älli

# WORDPRESS-TEEMAN SUUNNITTELU JA TOTEUTUS CASE: PUHELINKLINIKKA.FI

Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa WordPress-sisällönhallintajärjestelmän teema Puhelinklinikka Oy:lle. Organisaatiolla oli vanhat sivut, joista haluttiin säilyttää väriteema ja rakenne. Uusia lisäyksiä, joita sivuille haluttiin, olivat selkeys, elävyys ja sosiaalisen median näkyvyys. Myös alisivut, kuten hinnasto, toimitusehdot ja yhteydenotto piti suunnitella uudelleen. Sisältöä oli vanhoilla sivuilla jo valmiiksi, mikä helpotti uuden ulkoasun suunnittelua.

Teoriaosaa kuvaa hieman WordPress-sisällönhallintajärjestelmä ja sen historia. Lisäksi se kertoo ohjelmointi-, merkintä- ja tyylikielistä, joita tarvitaan, kun WordPress-teema suunnitellaan ja toteutetaan alusta alkaen. Teemojen toteuttamisessa käytetään WordPressin rajapintaa, johon tässä opinnäytetyössä keskitytään. Opinnäytetyö sisältää myös muutamia koodinpätkiä, joita esiintyy WordPress-teemojen rakennetiedostoissa. Koodinpätkät sisältävät PHP-ohjelmointikieltä ja HTML-merkintäkieltä.

Web-sivujen ulkoasun oli tarkoitus saada vahvasta tekstipainotteisesta web-sivusta hieman elävämmäksi käyttämällä enemmän kuvia, värejä ja hieman animaatioita. Projektin aloitettiin kesällä 2012 ja se kesti noin kolme kuukautta. Lopputulos oli toimeksiantajan mieleen ja myös Puhelinklinikka Oy:n Facebook-sivun kautta on tullut positiivista palautetta.

## ASIASANAT:

WordPress, julkaisujärjestelmä, ohjelmointirajapinta, teemat, HTML5, CSS3, PHP

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | e-Business Systems

December 2013 | 25 pages

Instructor: Minna Paakki

Aleksi Älli

# DESIGNING AND CREATING A WORDPRESS THEME FOR PUHELINKLINIKKA.FI

The aim to this thesis was to design and create a WordPress content management systems theme to Puhelinklinikka Oy. The company already had a website, the color theme and the structure of which it wished to keep. The new and wanted features to the new site included clarity, vividness and better visibility for social media. Subpages like price list, conditions and contact should also be redesigned. The contents from previous website helped with layout design process.

The theory first introduces the WordPress content management system and its history and follows by the programming, marking and styling languages which are needed when WordPress theme is designed from scratch. WordPress API is used when WordPress themes are being created and this is the focus in this study. The report also includes a few pieces of code which occur in the structure files. The codes include PHP programming language and HTML markup.

Web sites layout was intended to swift from heavily text weighted web site to more vivid one by using more pictures, color and a bit of animations. The project started in spring 2012 and it lasted about three months. The final result pleased the client and also got positive feedback in Puhelinklinikka's Facebook page.

## KEYWORDS:

WordPress, CMS, API, themes, HTML5, CSS3, PHP

# SISÄLTÖ

<b>KÄYTETYT LYHENTEET JA SANASTO</b>	<b>6</b>
<b>1 JOHDANTO</b>	<b>7</b>
<b>2 WORDPRESS</b>	<b>8</b>
2.1 Historia	8
2.2 Vaatimukset ja käyttöönotto	9
2.3 Ohjelmointikielet	9
2.3.1 HTML5	9
2.3.2 CSS3	10
2.3.3 PHP	10
2.4 The Loop	11
2.5 Rakenne	12
2.6 Teeman luonti	14
<b>3 WWW-VERKKOSIVUJEN TOTEUTUS, CASE: PUHELINKLINIKKA.FI</b>	<b>16</b>
3.1 Vaatimukset	16
3.2 Rakenne	16
3.2.1 Etusivu	17
3.2.2 Hinnasto	17
3.3 Ulkoasu	19
3.4 WordPress lisäosat	20
<b>4 POHDINTA</b>	<b>23</b>
<b>LÄHTEET</b>	<b>25</b>

## KUVAT

Kuva 1. Esimerkki The Loopin käytöstä. ....	11
Kuva 2. Teeman metatiedot CSS-tiedostossa.....	12
Kuva 3. WordPress-teeman metatiedot hallintasivulla. ....	12
Kuva 4. Yksinkertaisen index.php:n sisältö. ....	13
Kuva 5. Yksinkertaisen header.php:n sisältö.....	14
Kuva 6. Etusivulla näkyvät ”oikotiet”.....	17
Kuva 7. Hinnasto tulostettuna sivuille.....	18

Kuva 8. Puhelinklinikka.fi:n uusi logo. ....	19
Kuva 9. Sivulla näkyvä diashow. ....	20
Kuva 10. Contact Form 7 muokattuna.....	21
Kuva 11. NextGEN Galleryn näkymä web-sivulla.....	22
Kuva 12. Tämänhetkinen ratkaisu hinnastossa.....	24
Kuva 13. Uusi ratkaisu hinnastoon.....	24

# KÄYTETYT LYHENTEET JA SANASTO

HTML	merkintäkieli, Hypertext Markup Language
CSS	tyylikieli, Cascading Style Sheets
PHP	ohjelmointikieli, PHP: Hypertext Preprocessor
Ohjelmointirajapinta	määritelmä, jonka mukaan ohjelmat voivat keskustella keskenään

# 1 JOHDANTO

Yhä useampi web-sivu rakennetaan käyttämällä sisällönhallintajärjestelmää julkaisualustana, jotta sen päivittäminen olisi helpompaa käyttäjälle. Jo web-sivujen suunnitteluvaiheessa otetaan huomioon sisällönhallintajärjestelmän antamat helpotukset, hyödyt ja mahdolliset ongelmat. Internetistä löytyy sivustoja, joihin teemojen kehittäjät voivat pistää kehittämiään teemoja myyntiin. Themeforest.com-sivuston kärkisijoilta löytyy WordPress-teema nimeltä Avada, jota on myyty yli 44 600 kertaa ja sen hinta on 55 dollaria.

Tässä opinnäytetyössä keskitytään WordPress teemojen rajapintaan ja siihen, mitä kaikkea tarvitaan, kun se rakennetaan tyhjästä. Opinnäytetyö aloitetaan kertomalla hieman WordPressistä, sen historiasta ja mitä kaikkea vaaditaan sen käyttöönottoon. WordPress rajapinta koostuu PHP-ohjelmointikielestä ja teemat rakennetaan käyttäen HTML-merkintäkieltä ja CSS-tyylikieltä. Tämän takia opinnäytetyössä käydään lyhyesti läpi HTML5, CSS3 ja PHP.

Opinnäytetyöhön kuuluu myös toimeksianto, jossa kehitetään Puhelinklinikka Oy:lle uudet web-sivut. Puhelinklinikka Oy on vuonna 2011 perustettu matkapuhelinhuoltoa harjoittava yritys, joka toimii Turussa. Yritys huoltaa muun muassa Applen, Nokian ja Samsungin matkapuhelimia. Yrityksen perustajana toimii Jussi Salonen.

Toimeksiannossa lähdettiin liikkeelle vaatimuksista, kuten mitä web-sivuilla olisi hyvä olla, mitä vanhoilta sivuilta voisi säilyttää ja mitä pitäisi tuoda lisää esille. Rakennetiedostoja tarkastellaan kehittämisen ja ulkoasusuunnittelun näkökulmasta. Erityisrakennesivut, kuten etusivu ja hinnasto käydään erikseen tarkemmin läpi.

## 2 WORDPRESS

WordPress on yksi suosituimmista avoimeen lähdekoodiin perustuva sisällönhallintajärjestelmistä, jota käytetään niin bloggaus- alustana kuin myös yritysten web-sivujen alustana. WordPress on rakennettu PHP-kielellä ja käyttää MySQL-relaatiotietokantaohjelmistoa tiedon tallentamista varten.

Muita WordPressin kaltaisia avoimen lähdekoodiin perustuvia sisällönhallintajärjestelmiä ovat esimerkiksi Joomla ja Drupal.

### 2.1 Historia

WordPress on vuodesta 2003 asti kehitetty sisällönhallintajärjestelmä, jonka pääkehittäjä on Matt Mullenweg. WordPress tunnettiin pitkään pelkkänä blogialustana ja vasta myöhemmin sitä alettiin käyttää erilaisten www-sivujen sisällönhallintajärjestelmänä. Michael Valdrighin kehittämä b2/cafelog, WordPressin edeltäjä, oli myös kirjoitettu PHP-koodilla. Tällöin Valdrigh oli vasta oppimassa PHP:ta, joten kokeneempien kehittäjien oli hankala lukea hänen tuottamaansa koodia, mutta se oli myös samalla helpompaa uusille kokemattomille kehittäjille (WordPress 2013a).

Vuonna 2005 WordPressin versio 1.5 toi mukanaan teemajärjestelmän, jossa teema oli jaettu eri tiedostoihin yleisempien www-sivujen elementtien perusteella, kuten ylätunniste (eng. header), alatunniste (eng. footer) ja sivupalkki (eng. sidebar). Tätä tekniikkaa käyttämällä käyttäjien oli helppo tehdä muutoksia ulkoasuun ja nähdä ne heti. Michael Heilemannin tekemä teema "Kubrick" oli WordPressin vakioteemana ja inspiroi yhteisöä suuresti kehittämään uusia teemoja, joiden vaihtaminen onnistui nyt yhdellä klikkauksella (WordPress 2005). WordPressin nykyinen versio 3.6, "Oscar", joka on nimetty jazzpianisti Oscar Petersonin mukaan, sisältää vakioteeman nimeltä "Twenty Thirteen", joka on täysin responsiivinen jokaiselle mobiili- ja taulutietokonelaitteelle (WordPress 2013b).



## 2.2 Vaatimukset ja käyttöönotto

Niin kuin mikä tahansa muukin sisällönhallintajärjestelmä, niin myös WordPress tarvitsee toimiakseen www-palvelimen, johon WordPress asennetaan ja missä www-sivut tulevat näkymään. Nykyinen WordPress versio 3.6 tarvitsee toimiakseen PHP-version 5.2.4 tai uudemman ja MySQL-version 5.0 tai uudemman. WordPress myös suosittelee palvelimen olemaan joko Apache tai Nginx. WordPress on myös listannut muutaman maksullisen www-palvelimen tarjoajan, jotka ovat valmiina WordPressin asennusta varten ja jota käyttäessään se myös tukee WordPress.orgia. WordPress myös tarjoaa ilmaista blogitilaa käyttäjille, jotka tyytyvät vähempään (WordPress 2013c).

WordPressillä on tunnettu ”viiden minuutin asennus”, joka tekee asennuksesta nopeaa ja helppoa. WordPressin päivittäminen on nykyisin myös helppoa automaattisen päivitystyökalun kanssa. WordPressiä asentaessa käyttäjällä pitää olla oikeudet www-palvelimelle, tekstieditori ja selain (WordPress Codex 2013a).

## 2.3 Ohjelmointikielet

WordPress teemat koostuvat pääosin kolmesta eri kielestä: HTML, CSS ja PHP. HTML on kuvauskieli ja sitä käytetään www-sivun kokoamiseen, jossa teksti, kuvat, videot ja lomakkeet yhdistyvät yhdeksi toimivaksi kokonaisuudeksi. CSS on tyylikieli, jolle määritetään tietyt tyyliohjeet, miten eri elementit www-sivulla näkyvät. PHP on ohjelmointikieli, jolla www-sivun eri toiminnot toteutetaan, kuten www-sivun hakutoiminto (Brazell 2011, 149).

### 2.3.1 HTML5

HTML tulee sanoista HyperText Markup Language ja se on vuodesta 1990 asti kehitetty kuvauskieli ja se tunnetaan erityisesti www-sivujen ohjelmointikielenä. Sen tarkoitus on kuvata www-sivujen rakenne ja toimia ihmisen ja tietokoneen

välisenä tulkkina, jotta tekstistä saadaan visuaalisesti luettava. (Cook 2012 ja Garber, 2–3).

HTML5 on sen uusin versio ja tuo mukanaan uudistuksia muun muassa HTML-kielen merkintätapaan. Ennen käytetyt DIV-elementin ID-attribuuttiin annetut arvot, kuten header, nav ja footer on HTML5:ssä korvattu tageilla <header>, <nav> ja <footer>, mikä selkeyttää koodin lukua (HTML5 2013).

### 2.3.2 CSS3

CSS (Cascading Style Sheet) on tyylikieli, jota käytetään www-sivujen tyyliohjeena. Kun HTML antaa sivulle rakenteen, niin CSS kertoo tyyliohjeensa kautta sen tyylimäärittelyt ja antaa sivulle ulkonäön.

CSS3 on uusin ja kehittynein versio CSS:stä ja tarjoaa uusia mahdollisuuksia www-sivujen kehityksessä. Isoimmat uudistukset CSS3:ssa verrattuna vanhempiin versioihin on liukuvärit, ”sulatetut” fontit, varjostukset, siirtymiset ja animaatiot. Tunnetuimpien selainten nykyversiot tukevat jo hyvin näitä kaikkia CSS3:n tuomia uudistuksia (Cook 2012 ja Garber, 9).

### 2.3.3 PHP

HTML on merkintäkieli, joka antaa sivuille rakenteen, ja CSS on tyylikieli, joka antaa sivuille ulkoasun. PHP puolestaan on ohjelmointikieli, jota käytetään, kun web-sivuista tehdään dynaamisia. PHP on dynaamisten web-palveluiden tuottamisessa johtava ohjelmointikieli ja vuonna 2013 sitä on käytetty jo yli 224 miljoonaan eri web-sivustoon (Netcraft 2013 & PHP 2013). WordPress koostuu PHP:sta ja sen osaaminen katsotaan eduksi, kun ollaan tekemisissä WordPress teeman kehityksen rajapinnan kanssa.

## 2.4 The Loop

The Loop on PHP-koodia ja WordPressin käyttämä päämekanismi, joka hakee artikkeleita (eng. posts) tietokannasta sivulle, joten se on tärkein osa teemojen rakennetiedostoista. Käyttäjä voi määritellä The Loopin hakemaan tietyn artikkelin, kategorian, sivun tai päivämäärän, mutta jos näitä ei ole määritelty, WordPress käyttää ennalta määriteltyjä asetuksia (Brazell 2011, 126).

```

1  <?php
2      /* Jos on olemassa esitettäviä artikkeleita */
3      if ( have_posts() ) :
4          /* Haetaan artikkeleita niin kauan kun niitä löytyy */
5          while ( have_posts() ) :
6              /* Haetaan artikkeli */
7              the_post();
8      ?>
9
10 <?php
11     /* Lopetetaan artikkeleiden haku */
12     endwhile;
13     /* Jos artikkeleita ei löytynyt */
14     else:
15 ?>
16 <!-- Luodaan kappale virhesanomaa varten käyttämällä HTML-merkkintäkieltä -->
17 <p>
18     <?php
19         /* Tulostetaan virhesanoma */
20         _e('Artikkeleita ei löytynyt.');

```

Kuva 1. Esimerkki The Loopin käytöstä.

Syy miksi PHP-ohjelmointikieltä käytetään näin pätkissä, johtuu siitä, että esimerkiksi teemojen rakennetiedostot sisältävät suurimmaksi osaksi HTML-merkkintäkieltä, joten PHP joudutaan upottamaan HTML:n sekaan. WordPress myös käyttää hakasulkujen sijaan kaksoispistettä, jolloin myös lopetuslauseke pitää löytyä, esimerkiksi endwhile ja endif (Brazell 2011, 127).

## 2.5 Rakenne

Teema voi koostua monesta eri PHP-tiedostosta, mutta tärkeimmät ja yleisimmät rakennetiedostot ovat index.php, header.php, footer.php, sidebar.php, page.php ja style.css. Näistä rakennetiedostoista WordPress vaatii style.css:n tunnistaakseen teeman. Style.css sisältää teeman metatiedot (kuva 2), joista WordPress hakee teemalle esimerkiksi nimen, www-osoitteen, version, tekijän ja version, joka näkyy WordPress-hallintasivulla (kuva 3). Hallintasivua varten teemalle voi myös asettaa kuvankaappauksen, jonka WordPress tunnistaa, kun se on png-kuvaformaattissa ja tiedostonimenä on screenshot (Brazell 2011, 150).

```

1  /*
2  Theme Name: Puhelinklinikka
3  Theme URI: puhelinklinikka.fi
4  Description: A WordPress theme designed for website puhelinklinikka.fi
5  Version: 2.0
6  Author: Aleks A. Älli
7  Author URI: http://aleksialli.fi
8  */

```

Kuva 2. Teeman metatiedot CSS-tiedostossa



Kuva 3. WordPress-teeman metatiedot hallintasivulla.

Index.php on tiedosto, joka yhdistää muut rakennetiedostot toisiinsa ja toimii yleensä etusivuna, kun kyseessä on blogi (kuva 4). Yksinkertaisessa index.php-rakennetiedostossa se sisältää vain WordPressin The Loopin ja sisällyttämis-

metodit, kuten `get_header()`, `get_sidebar()` ja `get_footer()`, jotka hakevat `header.php`-, `sidebar.php`- ja `footer.php`-tiedostojen sisällöt (Brazell 2011, 150).

```

1  <?php
2      /* Haetaan teeman header.php -rakennetiedoston sisältö ja aloitetaan The Loop */
3      get_header();
4      if( have_posts() ) :
5          while( have_posts() ) : the_post();
6      ?>
7
8      <!-- Asetetaan sivuille H2-otsikko, jonka sisään upotetaan
9      WordPress metodi joka, hakee artikkelin otsikon. Julkaisuaika
10     ja julkaisija upotetaan em-tagin sisälle joka kursivoi tekstiä. -->
11     <h2><a href=?php the_permalink(); ?>><?php the_title(); ?></a></h2>
12     <em>Julkaistu <?php the_time(); ?> Kirjoittanut <?php the_author(); ?></em>
13
14
15     <?php
16         /* Haetaan ja tulostetaan artikkelin sisältö. */
17         the_content();
18     ?>
19
20
21     <?php
22         /* Haetaan comments.php -rakennetiedoston sisältö */
23         comments_template();
24     ?>
25
26
27     <?php
28         endwhile;
29         else :
30     ?>
31
32     <h2>Artikkelia ei löytynyt.</h2>
33     <p>Kyseistä artikkelia ei löydy.</p>
34
35     <?php
36         /* Lopetetaan The Loop ja haetaan sidebar.php ja footer.php -rakennetiedostot. */
37         endif;
38         get_sidebar();
39         get_footer();
40     ?>

```

Kuva 4. Yksinkertaisen `index.php`:n sisältö.

Header.php tiedosto on nimensä mukaan WordPress teeman ylätunniste eli siihen kerätään yleensä kaikki web-sivun ylälaidassa olevat elementit, kuten logo ja navigaatio (kuva 5). Ylätunnisteesta löytyvät myös sivun CSS-tiedoston sijainti, metatiedot, RSS- tai ATOM-syötteiden sijainti, joiden tarkoitus on samalla välittyä koko teemalle (Brazell 2011, 153) .

```

1 <?php
2 /**
3  * Header.php
4  */
5 ?>
6
7 <!DOCTYPE html>
8 <head>
9     <!-- Haetaan teeman asetuksista teemalle merkistö, otsikko ja kuvaus -->
10     <meta charset="<?php bloginfo('charset'); ?>">
11     <title><?php bloginfo('name'); ?></title>
12     <meta name="description" content="<?php bloginfo('description'); ?>" />
13
14     <!-- Haetaan teeman rakennetiedoista CSS-tyylitiedostot -->
15     <link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/normalize.css">
16     <link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/style.css">
17
18 </head>
19 <body>
20     <header>
21         <div>
22             <!-- Haetaan teeman rakennetiedoista sivujen logo ja annetaan sille linkiksi sivujen osoite -->
23             <a href="<?php bloginfo('url'); ?>" class="logo">
24                 
25             </a>
26         </div>
27     </header>
28     <!-- Haetaan teeman navigaatio, joka on määritelty hallintasivun kautta -->
29     <nav>
30         <?php wp_nav_menu(); ?>
31     </nav>
32
33 <!-- Header.php -tiedoston loppu -->

```

Kuva 5. Yksinkertaisen header.php:n sisältö.

Footer.php on kuten header.php, mutta se toimii sivun alatunnisteena. Web-sivujen alatunnisteeseen on yleensä tapana pistää tekijänoikeusmerkinnät, yhteystiedot ja web-sivujen sivukartta (Brazell 2011, 154).

Sidebar.php on myös yksi yleisimmästä teeman osista ja yleensä sitä käytetään web-sivun oikealla tai vasemmalla puolella, mutta sen nimestä huolimatta, sitä voidaan käyttää muuallakin kuin sivupalkkina. Nykyään sidebar.php tehdään teeman widgetiksi, joten sivujen hallinnoija voi hallintapaneelin kautta muokata sitä tarpeekseen tai ottaa kokonaan pois (Brazell 2011, 155).

## 2.6 Teeman luonti

WordPress teeman luonti aloitetaan yleensä tekemällä yksi iso staattinen HTML-sivu, johon on tarkoitus saada suunnitellun ulkoasun rakenne kasattua. Tästä isosta staattisesta HTML-sivusta saadaan teeman rakennetiedoista header.php, footer.php, sidebar.php ja usein myös etusivun oma erityismallisivu rakenteeltaan valmiiksi. Myös CSS-tyylitiedosto luodaan tässä vaiheessa ja lin-

kitetään HTML-sivulle. Jos teema luodaan ilman palvelinta, esimerkiksi kotikoneelle, on sille oltava PHP-tuki. PHP:n tuki on välttämätöntä myös WordPressin kanssa, joten on hyvä asentaa kotikoneelle kevyt palvelin käyttämällä esimerkiksi WAMP-kehitysympäristöä, jonka avulla koko teeman toteutuksen voi ensin tehdä turvallisesti ja nopeasti ensin kotikoneella suljetussa ympäristössä ja sen jälkeen pakata sen ja lähettää ulkoiselle www-palvelimelle (McCollin ja Silver 2013, 23–24).

Valmiin staattisen HTML-sivun kääntäminen WordPress teemaksi tapahtuu jakamalla se eri rakennetiedostoihin ja ohjaamalla CSS-tiedostot, kuvat ja muut tiedostopolut rakennekansioihin käyttämällä WordPressin rajapintaa hyväksi. Tiedostopolut voivat palvelimesta riippuen olla erilaisia, mutta käyttämällä WordPressin rajapintaa, se ymmärtää aina ohjata ne suoraan rakennetiedostoihin, joten kun ne on kerran määritelty, niitä ei tarvitse enää vaihtaa ulkoiselle palvelimelle siirron jälkeen (McCollin ja Silver 2013, 85).

### **3 WWW-VERKKOSIVUJEN TOTEUTUS, CASE: PUHELINKLINIKKA.FI**

#### **3.1 Vaatimukset**

Toimeksiantaja antoi www-sivujen yleisilmeelle ns. ”vapaat kädet”, mutta väriteemaan olisi kuitenkin oltava sama kuin nykyisillä sivuilla eli valko-puna-musta. Toiveina olisi saada lisää elävyyttä ja kiinnostusta www-sivuille, joten graafista ulkonäköä pitäisi olla enemmän.

Sisällön puolesta www-sivujen etusivulle pitäisi saada uutiset paremmin esille ja samalla myös nykyisten sosiaalisten medioiden, kuten Facebookin, pitäisi näkyä paremmin.

Www-sivuilta löytyvä hinnasto on pelkästään kuvaformaattissa, joten se lisää sivujen latausaikaa ja sen päivittäminen on hankalaa. Tämä tulisi muuttaa WordPress sisällönhallintajärjestelmän kautta muokattavaksi, niin että käyttäjä voi itse lisätä hinnastoon uuden puhelinvalmistajan ja sille mallit. Myös sivuilta löytyvä yhteydenottolomakkeelle pitäisi tehdä parannuksia, kuten lisätä liitetiedostoja.

#### **3.2 Rakenne**

Puhelinklinikka Oy:n www-sivujen teema koostuu kolmesta eri rakennesivusta: header.php, page.php, footer.php ja muutama erityismallisivu, kuten hinnasto.php ja etusivu.php. Header.php ja footer.php ovat sivujen ylä- ja alatunnisteet, eli ne ovat jokaisella alisivulla samannäköiset kuin myös etusivulla. Sisällön tulostuksen näkymä vaihtelee jonkin sivun alasuilla, kuten ”Hinnasto” ja ”Yhteystiedot”, joissa sisällön tulostus on pelkkään tekstin tulostamiseen verrattuna hieman monimutkaisempaa.



### 3.2.1 Etusivu

Etusivulle oli tarkoitus saada kolme tärkeintä asiaa: nopea esittely ja tervetulo-toivotus, uusimmat uutiset ja houkuttelevaisuus. Tervetulopuhe ja esittely on tarkoitus olla ensimmäinen, minkä kävijä lukee sivuille tullessaan. Esittely on nopea ja antaa käyttäjälle heti selon siihen, mitä yritys tekee ja missä päin yritys sijaitsee. Esittelyn jälkeen asiakasta kehoitetaan tekemään kustannusarvio jo tänään, jonka jälkeen seuraa linkki mikä vie sivulle ”Yritys”, missä kerrotaan tarkemmin tietoa itse yrityksestä ja sen taustalla toimivasta henkilöstä ja hänen ammattitaidostaan.

Esittelyn viereen on koottu uutiset, johon WordPress hakee aina kolmen uusimman uutisen otsikkoa uutisarkistosta. Otsikkoa klikkaamalla sivu ohjautuu klikatun uutisen kohdalta uutisarkistoon, josta löytyy koko uutinen. Uutisarkistoon pääsee myös klikkaamalla etusivulla linkkiä ”Lisää uutisia”.

Etusivun alaosa löytyy kolme ns. ”oikotietä” (kuva 6), joiden on tarkoitus antaa kävijälle suunta sille, mitä tämä on etsimässä sivustolta: ”Huolto”, ”Kastunut puhelin” ja ”Kustomointi”. Jokainen ”oikotie” koostuu sitä kuvaavasta ikonista, lyhyestä tekstistä ja linkistä joka, ohjaa kävijän eteenpäin.



Kuva 6. Etusivulla näkyvät ”oikotiet”.

### 3.2.2 Hinnasto

Vanhojen sivujen hinnasto oli toteutettu käyttämällä kuvia ja niiden muokkaamiseen vaadittiin Adobe Photoshop -ohjelmisto, mikä teki hinnaston päivittämisestä työlästä ja hieman hankalampaa. Uusi hinnasto-sivu tehtiin dynaamiseksi

käyttämällä pelkästään WordPressin omaa ohjelmointirajapintaa, jolloin käyttäjät voivat itse lisätä hinnastoon uusia puhelinvalmistajamerkkejä, niiden malleja ja niiden korjaustapahtuman hintoja.

Hinnaston toiminnallisuus perustuu WordPressin käyttämiin artikkeleihin ja niiden kategorioihin. Kategoriat kuvaavat puhelimen merkkiä ja artikkeli sen mallia. Jos käyttäjä haluaa lisätä uuden puhelinvalmistajamerkin hinnastoon, hän luo ensin uuden kategorian, esimerkiksi Samsung, ja tämän jälkeen artikkeleita juuri luodun kategorian alle, kuten Galaxy S4. Eri korjaustapahtumat kullekin mallille luodaan artikkelin sisältä, josta löytyy avainkentät "palvelu1, palvelu1\_hinta, palvelu2, palvelu2\_hinta" jne. Kenttään "palvelu1" käyttäjä lisää korjaustapahtuman nimen, kuten "Näytön vaihto", ja "palvelu1\_hinta" korjaustapahtuman hinnan, kuten "139 €". Oletuksena jokaiselle mallille on annettu neljä korjaustapahtumaa, mutta käyttäjä voi kätevästi lisätä niitä käyttämällä samaa kaavaa eli kasvattaa avainkentissä näkyvän nimen numeroa.

Kun tarvittavat lisäykset tai muutokset on tehty ja tallennettu, niin WordPress hakee kaiken "Hinnasto"-kategorian alta ja tulostaa ne sivuille muotoiltuna (kuva 7).

## Apple

iPhone 4		iPhone 4S		iPhone 5	
Näytön vaihto	139€	Näytön vaihto	149€	Näytön vaihto	229€
Takalasin vaihto	59€	Takalasin vaihto	69€	Takakuoren vaihto	kysy
Näytön + takalasin vaihto	179€	Näytön + takalasin vaihto	179€	Näytön + takakuoren vaihto	kysy
Virtakytkimen vaihto	59€	Virtakytkimen vaihto	69€	Virtakytkimen vaihto	89€
Akun vaihto	59€	Akun vaihto	69€	Akun vaihto	79€
Kotinäppäimen vaihto	59€	Kotinäppäimen vaihto	69€	Kotinäppäimen vaihto	59€
Latausliittimen vaihto	59€	Latausliittimen vaihto	69€	Latausliittimen vaihto	89€

Kuva 7. Hinnasto tulostettuna sivuille.

### 3.3 Ulkoasu

Ulkoasun suunnittelu alkoi käyttämällä Adobe Photoshopia, jossa dokumentille annettiin alkumitoiksi 1200 pikseliä leveyttä ja 1200 pikseliä pituutta. Tämän jälkeen dokumenttiin asetettiin ruudukko, jotta ulkoasuun asetetut elementit pysyisivät linjassa. Ylätunniste koostuu kolmesta eri elementistä: logo, navigaatio ja kuvaslider, joka erotetaan logosta ja navigaatiosta eri värisellä taustalla. Logo on kaksivärinen teksti, jota seuraa kuva puhelimesta, jonka sisällä on sydän, jossa on nollaviiva. Vanhassa logossa esiintyi punainen risti valkealla pohjalla, joka aiheutti konfliktin, koska se on humanitaarinen suojamerkki, joten se piti korvata uudella logolla. Logo toteutettiin Adobe Illustrator -ohjelmistossa, jotta siitä saatiin vektori, joka mahdollistaa logon pysyvän täydessä kuvalaadussaan, koosta riippumatta. Logon alapuolelle myös lisättiin Puhelinklinikan käyttämä tunnuslause: ”Osaavaa matkapuhelinhuoltoa tois puol jokke”. Tämän jälkeen vektori tuotiin Illustratorista Photoshopiin ja asetettiin ylätunnisteeseen (kuva 8).



Kuva 8. Puhelinklinikka.fi:n uusi logo.

Navigaatiolinkkeihin laitettiin vain www-sivun isoimmat ja tärkeimmät sivut, loput sivut löytyvät alatunnisteen sivukartasta. Kuvasliderin (kuva 9) tarkoitus oli antaa sivulle eloa ja samalla esitellä korjattuja puhelimia ja johtaa sivulla kävijät kuvagalleriaan, josta löytyy lisää kuvia korjatuista ja kustomoiduista puhelimista.



Kuva 9. Sivuilla näkyvä diashow.

Ylätunnisteen jälkeen suunniteltiin sivun sisältöosuus. Etusivusta tehtiin muihin alasivuihin verrattuna vähän erilainen, koska siitä tulisi huomattavasti täyteläisempi kuin muista. Etusivun sisältöosuuden taustasta tehtiin kaksivärinen. Tervetulotoivotus ja uutiset asetettiin vierekkäin hieman tummemmalle taustalle, kun taas pikalinkit aseteltiin valkoiselle taustalle. Jokaista pikalinkkiä kuvaa niille tehdyt ikonit ja niiden tarkoitus on herättää kiinnostusta sivulla kävijälle ja ohjata häntä eteenpäin.

Sivujen alatunniste jaettiin kolmeen osaan: sivukartta, Facebook ja yhteystiedot. Sivukartta on hyvä laittaa alatunnisteeseen silloin kun, sivuilla on useampi alasivu, joka ei suoraan löydy navigaatiopalkista. Facebook haluttiin selvästi näkyville, joten se asetettiin isokokoisesti alatunnisteen, joka mahdollistaa sen näkyvyyden myös jokaisella alasivulla. Yhteystiedoista löytyy nopeat kontaktiyhteydet ja linkki yhteydenottolomakkeelle.

### 3.4 WordPress lisäosat

Rakenteen puolesta teeman hyväksi käytettiin kahta WordPress lisäosaa: "Contact Form 7" ja "NextGen Gallery". Contact Form 7 on yhteydenottolomaketta varten käytettävä lisäosa, jota on helppo muokata ja turvallinen. Contact Form 7 sisältää myös mahdollisuuden liitää tiedostoille, mikä on tärkeä ominaisuus yhteydenottolomakkeelle (kuva 10).

## Yhteydenottolomake

Etunimi	Sukunimi
<input type="text"/>	<input type="text"/>
Sähköposti (täytä huolellisesti!)	Paikkakunta
<input type="text"/>	<input type="text"/>
Puhelimen merkki	Puhelimen malli
<input type="text" value="---"/>	<input type="text"/>
Puhelimen väri	
<input type="text" value="---"/>	
Vikakuvaus tai muu viesti	
<input type="text"/>	
Liitetiedosto	
<input type="button" value="Valitse tiedosto"/> <input type="button" value="Ei valittua tiedostoa"/>	
<input type="button" value="LÄHETÄ LOMAKE"/>	

Kuva 10. Contact Form 7 muokattuna

Kuvagalleriaa käytetään lähinnä korjattujen puhelimien esittelyä varten ja tähän valittiin WordPress lisäosa NextGen Gallery, josta löytyi kaikki tarvittavat ominaisuudet. Kun siihen on laitettu tarvittavat asetukset, voi sitä käyttää kuka tahansa käyttäjä. NextGen Gallery osaa myös säädellä kuvan kokoa selainikkunan koon mukaan, joten se on täysin responsiivinen (kuva 11).

## Kuvagalleria



Kuva 11. NextGEN Galleryn näkymä web-sivulla.

## 4 POHDINTA

Pienellä WordPressin kokemuksella oli melko helppoa päästä kiinni sen rajapintaan ja teemojen kehittämiseen. WordPressin oma sivusto, [codex.wordpress.org](https://codex.wordpress.org), tarjoaa kattavan manuaalin ja WordPressin suuri yhteisö on myös hyvä keino pyytää apua vaativimpiin ongelmiin. Jos projektille asetetaan pitkä deadline tai sitä ei ole, pyrin käyttämään mahdollisimman paljon aikaa, jotta saisin parasta jälkeä. Tämän projektin aikana tein useita versioita ulkoasusta ja lähetin niitä toimeksiantajalle, joka kertoi, mitä voisi mahdollisesti muuttaa, poistaa tai lisätä, minkä jälkeen tein taas uuden version. Koko projektin keskustelu käytiin sähköpostin välityksellä ja se toimi todella hyvin.

Sivujen jatkokehityksessä kannattaa panostaa vielä enemmän sivun latautumisnopeuteen ja yleisesti sen toimivuuteen. Vaikka sivuista karsittiin paljon kuvia pois, voisi sitä vieläkin enemmän toteuttaa hyödyntämällä CSS3:n uusia ominaisuuksia. Esimerkiksi ylätunnisteen taustan voisi toteuttaa CSS3:n liukuväri-attribuutilla, "linear-gradient". Myös sivuilla esiintyvät ympyränmuotoiset ikonit voisi kaikki sijoittaa yhteen kuvaan, jota kutsutaan sprite-kuvaksi. Tämän jälkeen ympyrätaustan voisi tehdä hyödyntämällä CSS3:n "border-radius" ominaisuutta.

Sivuilta löytyvä hinnasto olisi myös hyvä rakentaa uudestaan hyödyntämällä WordPressin "custom post type"-metodia, jolloin artikkelikategorian sijaan, WordPress Loop hakee määritetyn post typen eli artikkelityypin. "Custom post type":ä varten pitää luoda ensin uusi artikkelityyppi teeman rakennetiedostoon nimeltä `functions.php`, jonka jälkeen WordPress tunnistaan uuden artikkelityypin hallintapaneelissa ja luo sille oman kohdan kun halutaan muokata hinnastoa. Tämän jälkeen `hinnasto.php`-mallisivulla oleva vanha Loop (kuva 12) muutetaan hakemaan artikkelit artikkelityypin mukaisesti (kuva 13).

```

1 <?php
2     $i = 0;
3     $cat_args=array(
4         'orderby' => 'name',
5         'order' => 'ASC',
6         'child_of' => '8' /* Hinnaston pääkategorian ID */
7     );
8     $categories=get_categories($cat_args);
9
10    /* Haetaan niin kauan kun alakategorioita löytyy */
11    while($i < sizeof($categories)):
12
13        /* Tulostetaan Hinnaston alakategorien(puhelinvalmistajien) nimet <header> -tagien väliin. */
14        echo "<header>".$categories[$i]->name."</header>";
15
16        endwhile;
17 ?>

```

Kuva 12. Tämänhetkinen ratkaisu hinnastossa.

```

1 <?php
2     $ottelut_args=array(
3         'orderby' => 'name',
4         'order' => 'ASC',
5         'post_type' => 'hinnasto' /* Hinnaston artikkelityypin nimi */
6     );
7     $puhelinvalmistajat = get_posts($puhelinvalmistajat_args);
8
9     /* Erotetaan jokainen valmistaja erilleen */
10    foreach ($puhelinvalmistajat as $puhelinvalmistaja) :
11        $puhelinvalmistajan_nimi = get_post_meta($puhelinvalmistaja->ID, 'puhelinvalmistajan_nimi', true);
12
13        /* Tulostetaan Hinnaston puhelinvalmistajien nimet <header> -tagien väliin. */
14        echo "<header>".$puhelinvalmistajan_nimi."</header>";
15
16    endforeach;
17 ?>

```

Kuva 13. Uusi ratkaisu hinnastoon.

Koska Puhelinklinikka on matkapuhelimien korjaamista ja kustomointia harjoittava yritys, niin myös sen sivut tulisi olla responsiiviset jokaiselle mobiili- ja tablettilaitteelle, riippumatta niiden käyttämästä resoluutiosta. Vaihtoehtona olisi sivujen uudelleen rakentaminen käyttämällä eri avoimeen lähdekoodiin perustuvia ohjelmistokehyksiä, kuten Jeet ja Gumby, jotka mahdollistavat sivun muodon muutoksen kävijän resoluution mukaiseksi. Toinen ratkaisu olisi rakentaa sivuista erikseen mobiiliversio, jolloin sivut tunnistaisivat, kun kävijä saapuu si-  
vuille ja ohjaisi kävijän mobiiliversioon.



## LÄHTEET

Brazell, A. 2011. WordPress Bible, second edition. Indianapolis: Wiley Publishing, Inc.

Cook, C. & Garber, J. 2012. Foundation HTML5 with CSS3. New York: Springer.

McCollin, R. & Silver, T. 2013. WordPress Theme Development Beginner's Guide, third edition. Birmingham: Packt Publishing Ltd.

Netcraft 2013. PHP just grows & grows. Viitattu 20.9.2013 <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>

HTML 2013. Wikipedia. Viitattu 10.10.2013 <http://fi.wikipedia.org/wiki/HTML>

PHP 2013. Wikipedia. Viitattu 10.10.2013 <http://fi.wikipedia.org/wiki/PHP>

WordPress 2005. News. Viitattu 28.8.2013 <http://wordpress.org/news/2005/02/strayhorn/>

WordPress 2013a. History. Viitattu 28.8.2013 <http://wordpress.org/about/history>

WordPress 2013b. News. Viitattu 29.8.2013 <http://wordpress.org/news/2013/08/oscar/>

WordPress 2013c. Requirements. Viitattu 29.8.2013 <http://wordpress.org/about/requirements/>

WordPress Codex 2013a. Installing WordPress. Viitattu 4.9.2013 [http://codex.wordpress.org/Installing\\_WordPress](http://codex.wordpress.org/Installing_WordPress)